**ИТМО Кафедра Информатики и прикладной математики**

Отчет по лабораторной работе №1

«Копирование файлов»

Выполнил: студент группы P3218

Гхази Даниэль

Преподаватель: Зыков А. Г.

2017 год

1. **Задание**

Создайте файл, содержащий массив структур, каждая из которых содержит информацию о Ваших результатах за три семестра (фамилия, семестр, дисциплина, оценка).

Реализуйте простое последовательное копирование содержимого файла тремя различными способами:

1. С использованием библиотеки С.

2. С использованием Windows.

3. С использованием вспомогательной функции Windows — CopyFile.

Результат вывести на печать рез-ты копирования и средний балл.

Для анализа способов копирования создайте произвольный файл большого размера (экспериментально увеличивая для получения наглядных результатов) и определите временные характеристики каждого из способов. Результаты выведите в виде таблицы.

Проведите сравнительный анализ способов. Достоинства и недостатки.

1. **Листинг программы**

#include "./Headers/Lab1Header.h"

int main()

{

char choise;

CHAR currentDirectory[BUFFERSIZE];

do

{

system("cls");

std::cout << " 1. C-Library function." << endl;

std::cout << " 2. Windows function." << endl;

std::cout << " 3. CopyFile function." << endl;

std::cout << " 4. Generate data files." << endl;

std::cout << " 5. All together." << endl;

std::cout << " ESC - Exit." << endl;

choise = \_getch();

clock\_t startTime = clock();

switch (choise)

{

case '1':

{

CLibFunc(".\\Data\\inputData.txt");

break;

}

case '2':

{

WinFunc(".\\Data\\inputData.txt");

break;

}

case '3':

{

SecWinFunc(".\\Data\\inputData.txt");

break;

}

case '4':

{

DataGenerator();

break;

}

case '5':

{

std::cout << "\'C\' library func | "

<< "\'Windows\' API func | "

<< "\'CopyFile\' func |" << endl;

for(int i = 0; i < 5; i++)

{

char inputFilesPath[30];

char numBuf[12];

ZeroMemory(inputFilesPath, 30);

inputFilesPath[0] = '.';

strcat(inputFilesPath, "\\Data\\inputData");

itoa(i, numBuf, 2);

strcat(inputFilesPath, numBuf);

strcat(inputFilesPath, ".txt");

clock\_t startTime = clock();

//time\_t cFuncSecondsStart = time(NULL);

CLibFunc(inputFilesPath);

clock\_t startWinTime = clock();

//time\_t cFuncSecondsEnd = time(NULL);

WinFunc(inputFilesPath);

clock\_t startSecTime = clock();

//time\_t winFuncSecondsEnd = time(NULL);

SecWinFunc(inputFilesPath);

clock\_t endTime = clock();

//time\_t secWinFuncSecondsEnd = time(NULL);

// Results of copying.

std::cout << (double)(startWinTime - startTime) / CLOCKS\_PER\_SEC << "\t\t | "

<< (double)(startSecTime - startWinTime) / CLOCKS\_PER\_SEC << " | "

<< (double)(endTime - startSecTime) / CLOCKS\_PER\_SEC << "\t | "

<< endl;

}

break;

}

case escape:

{

return 0;

}

default:

{

continue;

}

}

clock\_t endTime = clock();

std::cout << "Total action time is: " << (double)(endTime - startTime) / CLOCKS\_PER\_SEC << " sec's." << endl;

int bytes = GetCurrentDirectory((DWORD)STRINGLENGTH, (LPSTR)currentDirectory);

std::cout << "Current directory is: " << currentDirectory << endl;

system("pause");

}

while (true);

system("pause");

return 0;

}

void DataGenerator()

{

FILE \*inputFile = NULL;

char inputFilesPath[24];

strcpy(inputFilesPath, ".\\Data\\inputData.txt");

char bufferForSurname[6];

char bufferForChar[12];

char bufferForNumb[1];

inputFile = fopen(inputFilesPath, "wb");

std::cout << "Output file's location is: " << inputFilesPath << endl;

for(int i = 0; i < 1000000; i++)

{

srand(time(NULL));

int semesterNum, mark;

char surname[6], lessonName[11];

strcpy(surname, "Ghazi ");

strcpy(bufferForSurname, surname);

fwrite(bufferForSurname, 1, 6, inputFile);

semesterNum = rand() % 2 + 1;

itoa(semesterNum, bufferForNumb, 10);

fwrite(bufferForNumb, 1, 1, inputFile);

strcpy(lessonName, " SomeLesson");

strcpy(bufferForChar, lessonName);

fwrite(bufferForChar, 1, 12, inputFile);

mark = rand() % 5 + 2;

itoa(mark, bufferForNumb, 10);

fwrite(bufferForNumb, 1, 1, inputFile);

strcpy(bufferForNumb, "\r\n");

fwrite(bufferForNumb, 1, 2, inputFile);

}

fclose(inputFile);

for(int i = 0; i < 5; i++)

{

char numBuf[1];

char inputFilesPath[30];

char bufferForChar[12];

ZeroMemory(inputFilesPath, 30);

inputFilesPath[0] = '.';

strcat(inputFilesPath, "\\Data\\inputData");

itoa(i, numBuf, 2);

strcat(inputFilesPath, numBuf);

strcat(inputFilesPath, ".txt");

inputFile = fopen(inputFilesPath, "wb");

std::cout << "Output file's location is: " << inputFilesPath << endl;

for(int j = 0; j < 1000000 \* (2 + i); j++)

{

srand(time(NULL));

int semesterNum, mark;

char surname[11], lessonName[11];

strcpy(surname, "Ghazi ");

strcpy(bufferForChar, surname);

fwrite(bufferForChar, 1, 11, inputFile);

semesterNum = rand() % 4 + 1;

itoa(semesterNum, bufferForNumb, 2);

fwrite(bufferForNumb, 1, 1, inputFile);

strcpy(lessonName, " SomeLesson");

strcpy(bufferForChar, lessonName);

fwrite(bufferForChar, 1, 12, inputFile);

mark = rand() % 5 + 2;

itoa(mark, bufferForNumb, 2);

fwrite(bufferForNumb, 1, 2, inputFile);

strcpy(bufferForNumb, "\r\n");

fwrite(bufferForNumb, 1, 2, inputFile);

}

fclose(inputFile);

}

}

void CLibFunc(char inputFilesPath[])

{

FILE \*inputFile = NULL, \*outputFile = NULL;

char outputFilesPath[] = ".\\Data\\outputData.txt";

char buffer[BUFFERSIZE];

size\_t inputBufferSize, outputBufferSize;

inputFile = fopen(inputFilesPath, "rb");

outputFile = fopen(outputFilesPath, "wb");

if (outputFile == NULL)

{

std::cout << "Error, cann't open destination file!" << endl;

system("pause");

return;

}

while (inputBufferSize = fread(buffer, 1, BUFFERSIZE, inputFile) > 0)

{

outputBufferSize = fwrite(buffer, 1, BUFFERSIZE, outputFile);

//std::cout << buffer << endl;

}

fclose(inputFile);

fclose(outputFile);

return;

}

void WinFunc(char inputPath[])

{

HANDLE inputHandle, outputHandle;

DWORD numberOfInput, numberOfOutput;

CHAR buffer[BUFFERSIZE];

LPCSTR inputFilesPath = (LPCSTR)inputPath;

LPCSTR outputFilesPath = ".\\Data\\outputData.txt";

inputHandle = CreateFile(inputFilesPath, GENERIC\_READ, 0, NULL, OPEN\_EXISTING, 0, NULL);

if (inputHandle == INVALID\_HANDLE\_VALUE)

{

std::cout << "Cann't open input file."<< endl;

return;

}

outputHandle = CreateFile(outputFilesPath, GENERIC\_WRITE, 0, NULL, CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (outputHandle == INVALID\_HANDLE\_VALUE)

{

std::cout << "Cann't open output file." << endl;

return;

}

while (ReadFile(inputHandle, buffer, BUFFERSIZE, &numberOfInput, NULL) && numberOfInput > 0)

{

WriteFile(outputHandle, buffer, numberOfInput, &numberOfOutput, NULL);

}

if (numberOfInput > numberOfOutput)

{

std::cout << "Writing error!" << endl;

system("pause");

return;

}

CloseHandle(inputHandle);

CloseHandle(outputHandle);

return;

}

void SecWinFunc(char inputPath[])

{

LPCSTR inputFilesPath = ".\\Data\\inputData.txt";

LPCSTR outputFilesPath = ".\\Data\\outputData.txt";

if (!CopyFile(inputFilesPath, outputFilesPath, FALSE))

{

std::cout << "Writing error!" << endl;

}

return;

}

1. **Вывод**

Таким образом, существует три способа копирования файлов в системе Windows: первый, использующий библиотеку языка С, может быть применен в том случае, если требуется возможность как можно более быстрой адаптации программы под альтернативные операционные системы, однако не предъявляется особых требований к производительности; второй способ, использующий функцию CopyFile, является наиболее эффективным в плане быстродействия. Наименее «приоритетным» в рассмотренных отношениях представляется способ, использующий Windows API, поскольку он как сильно проигрывает по времени работы, так и не предоставляет достаточной переносимости программ, хотя и обеспечивает высокую гибкость, если программа запускается только в рамках Windows.